19

Popular Computing

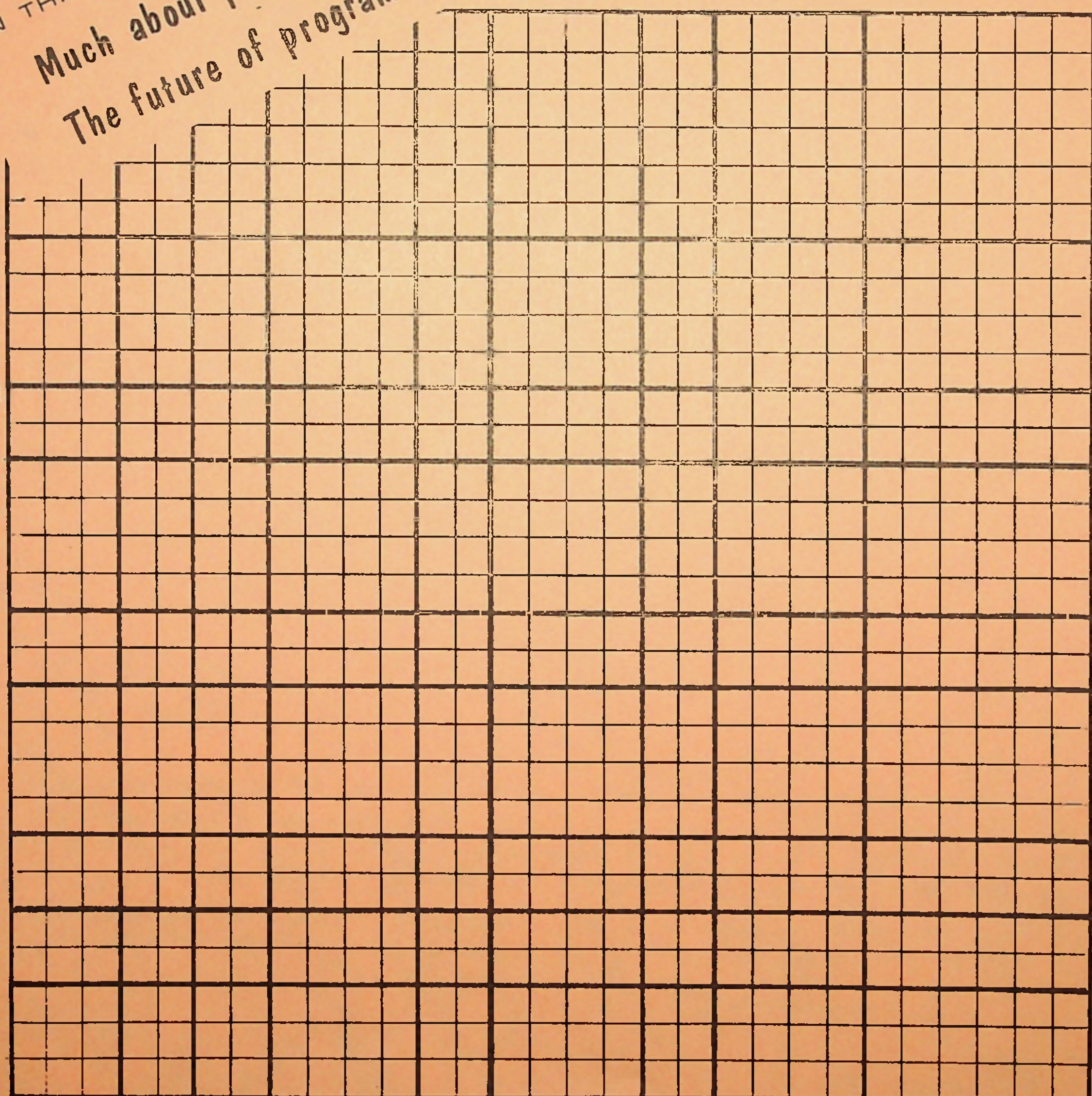
VOL 2 NO 10

OCTOBER 74

IN THIS ISSUE:

Much about primes

The future of programmers



The Primes Lattice

The figure on the cover shows a lattice in which the heavy lines are the coordinate lines for the odd primes (3, 5, 7, 11, 13, 17, 19, 23, 29). Most of the areas enclosed by the heavy lines are rectangles, but some areas are squares. In the table given here:

5	13	25	.520000
7	25	49	.510204
11	41	121	.338843
13	61	169	.360947
17	109	289	.377163
19	137	361	.379501
23	217	529	.410208
29	253	841	.300832

the first column shows the limit of the lattice to be considered; the second column gives the area within that limit that is enclosed by squares; the third column shows the total area up to the limit; and the fourth column shows the ratio of the area enclosed by squares to the total area. Thus, up to the limit 29, there are 16 2x2 squares, 9 4x4 squares, and one each of 3x3 and 6x6, for a total of 253 square units out of 841.

As the limits are increased, will the ratio stabilize? □

POPULAR COMPUTING is published monthly at Box 272, Calabasas, California 91302. Subscription rate in the United States is $18 per year, or $15 if remittance accompanies the order. For Canada and Mexico, add $4 to the above rates. For all other countries, add $6 to the above rates. Back issues $2 each. Subscriptions may begin with any issue.

Many intriguing problems in numbers involve the testing of a given number for primality. The most straightforward method for determining whether or not a number N is prime is to divide N (odd) by every odd prime less than or equal to the square root of N.

D. H. Lehmer, in a letter to Thomas R. Parkin (May 18, 1965) suggested the following improvement:

"When a number N is being tested for primality and fails to be divisible by small primes, say less than L, then give up the search which may cost on the order of $\sqrt{N}$ units of effort and apply the exponential test

$$2^N \not\equiv 2 \bmod N \qquad (A)$$

(which implies that N is composite) whose cost is only of the order of log N. In case (A) fails to hold, we don't know for sure that N is a prime, but then you can always pick up the old program at L and continue as before.

"Come to think of it, perhaps one should replace L by L+d; it would be a shame to quit at L and have the factor just around the corner!"

For most primes, $2^N \equiv 2 \bmod N$, but there are infinitely many composite numbers for which the relation holds, although such numbers are relatively rare. For example,

$$2^{561} \equiv 2 \bmod 561$$

and other examples are 645, 4371, 10585, 88561, and 137149. Oliver Gross has shown that if p is a prime greater than 3, then

$$N = (2^p - 1)\left(\frac{2^p + 1}{3}\right)$$

is always such a number. The generation of the first 7 such numbers is shown here.

p	2^p	$2^p - 1$	$\frac{2^p + 1}{3}$	N
5	32	31	11	341
7	128	127	43	5461
11	2048	2047	683	1398101
13	8192	8191	2731	22369621
17	131072	131071	43691	5726623061
19	524288	524287	174763	91625968981
23	8388608	8388607	2796203	23456248059221

Searching for Primes

PC19-3

Lehmer's scheme is one of many ingenious devices to make tractable the problem of determining primality of numbers that are large (in the sense of lying beyond the range of published tables. Systematic tables of the primes up to 104,000,000, and scattered tables in higher ranges, are available.) But even so, the value of "large" soon becomes excessive. For a number like

$$N = 2722258935367507707706996859454145691647$$

(which is $2^{131} - 1$), using the straightforward method of dividing by every prime less than or equal to the square root of N, and assuming that

1) all the base primes are known and available (there are some $2.6 \cdot 10^{18}$ of them), and

2) division of N by any of the base primes can be accomplished in one microsecond,

then the task of verifying a number of the size of N as prime or composite could take 85000 years. Clearly, for numbers larger than this N, the task of determining primality is fairly hopeless without special techniques.

Just such a technique is furnished by the Lucas-Lehmer test for numbers of the form

$$M = 2^p - 1,$$

the so-called Mersenne numbers. In 1644, Father Marin Mersenne conjectured that M is prime for only four values of p above 19; namely, for p = 31, 61, 127, and 257. For p = 257, it was soon established that M is composite. Up to 1964, the values of p for which M is known to be prime are these:

2	19	521	4253	
3	31	607	4423	(Selfridge, 1961)
5	61	1279	9689	(Gillies, 1963)
7	89	2203	9941	(Gillies, 1963)
13	107	2281	11213	(Gillies, 1963)
17	127	3217		

The values to the right of the line were all verified by computer, starting in 1952. (As late as 1950, attempts to find a prime larger than $2^{127} - 1$ ended in failure. The number $3 \cdot 2^{159} + 1$ had been suggested as a likely candidate, but it turned out to be composite. The report of the research establishing that fact concludes "Thus another attempt to discover a larger prime...ends in disappointment.")

The Lucas-Lehmer test proceeds as follows. Start with $A_1 = 4$. Calculate

$$A_n = \left[(A_{n-1})^2 - 2\right] \bmod M$$

M is prime if and only if $A_{p-1} = 0$. The complete tests for p = 11 and 13 are shown here:

$M = 2^{11} - 1$ Step	{Square - 2}	Mod M
1		4
2	14	14
3	194	194
4	37634	788
5	620942	701
6	491399	119
7	14159	1877
8	3523127	240
9	57598	282
10	79522	1736
	not prime	

$M = 2^{13} - 1$ Step	{Square - 2}	Mod M
1		4
2	14	14
3	194	194
4	37634	4870
5	23716898	3953
6	15626207	5970
7	35640898	1857
8	3448447	36
9	1294	1294
10	1674434	3470
11	12040898	128
12	16382	0
		prime

The Lucas-Lehmer test is cheap to perform on a computer, compared to any known test for arbitrary numbers. Nevertheless, the amount of computation increases rapidly with p, and Gillies' results stood as the records for 8 years. Bryant Tuckerman, of IBM's Thomas J. Watson Research Center, established the number $M = 2^{19937} - 1$ as the 24th Mersenne prime on March 4, 1971, using a 360/91 and 40 minutes of CPU time. His description of this work is found in the Proceedings of the National Academy of Science, Volume 68, No. 10, October 1971. A printout of the decimal expansion of the largest known prime is shown. Mr. Tuckerman has applied the test to all values of p less than 20000.

The bulk of the work in the Lucas-Lehmer test lies in the squaring of a p-bit number (in the case of the 24th Mersenne number, the worst case is the squaring of a number of 19937 bits). The reduction modulo M (since M is always an unbroken string of 1 bits) can be done by subtracting and adding single 1's. For example, in the test for M_{11}, stage 5 produces the number 620942. The reduction modulo M = 2047 is as follows in binary:

Text continued on page 7

THE 24TH MERSENNE PRIME IS $2^{19937} -1$ =

4315424	79738	81626	48055	23551	63379	19839	05393	50432	26711	50516	52505	41403	33068	01376	58091	13045	13629	31858	46655
45269	93825	76488	35317	90221	73345	84413	90952	82691	54609	16801	90078	75343	74139	62968	01920	11448	64809	02661	41431
84432	76980	30006	67281	04984	09545	15881	76077	13296	98437	62134	62179	03963	91341	28520	56276	19600	51310	66463	76648
61599	42366	75486	53748	02419	64350	29593	51686	62363	90904	79483	47692	31397	83013	77820	78571	24190	54474	33284	45291
83172	97324	23108	88265	08132	16264	69451	07770	78122	82829	44477	50226	80488	05782	00287	64659	39916	47662	65200	90056
14958	00344	05435	36903	89862	89406	17928	72011	12083	36148	08447	48291	35473	28367	27787	95656	48307	84690	91169	45866
23016	97024	01260	24018	70287	46650	03344	57745	70315	43129	29960	25187	78079	01193	75902	86317	10841	49642	47337	89862
67503	30896	13749	05766	34090	52895	72290	01603	80005	71630	87519	13739	79555	04746	81543	33253	47499	10462	48132	50451
63417	96551	47057	54814	59200	85947	26148	36213	87555	71168	64445	78975	08862	77996	48730	43084	50484	22342	06292	66518
55602	43393	39190	84436	89210	18424	84467	70427	27664	60185	29149	25277	28092	26975	38426	77025	73339	28954	40120	54658
95610	34765	88553	86633	90254	62899	62132	64328	24257	48035	78623	35806	08154	69654	69325	63833	32767	07698	99439	77488
85266	87278	52745	10029	63059	14696	38757	15425	73553	44759	79734	46310	06783	67393	32740	21499	30968	77829	67413	91514
59960	23742	13629	89872	06114	31410	40214	72389	98090	96281	89158	90645	69393	44833	30994	16963	22958	77995	84899	33667
47014	87176	34948	05549	99616	30515	41225	40346	52970	07721	14623	13557	04081	49309	86630	65733	67719	11728	53987	09574
81678	16256	08421	28233	80168	62533	45864	31254	03467	08061	35273	54327	07144	78876	86186	19833	20777	28064	48066	91125
71319	72625	81763	15131	35964	29547	76357	63678	37019	34983	51784	62144	29496	07571	90918	05462	51141	43666	38418	94338
52576	45228	93476	52454	63153	57404	68786	22894	58856	54608	56205	80424	68987	37243	69214	45092	31537	76984	07168	19837
65382	37748	61419	62070	41548	10637	93651	23192	81799	90066	21766	46716	71134	71632	71548	17958	77005	38269	43934	00403
06170	04576	91135	34918	78748	88923	42934	93401	45170	57171	61811	25795	88888	92774	95426	97714	99145	49623	91639	40148
22985	02533	16515	11431	27880	20090	56808	45650	68188	77266	60983	16368	83884	90562	18222	62933	98654	86456	69080	67219
17047	40408	89134	98356	85662	42806	32311	98520	43682	63294	15290	75297	27983	43429	44650	99922	06368	78136	71540	91702
65577	27273	91329	42427	75293	49082	60058	58847	66523	15095	74170	77831	91001	61684	75685	65867	31928	60882	07017	97603
07269	84998	73548	36042	37173	46602	57694	34723	55063	01744	11887	41412	92438	95814	15491	00609	75221	68822	30887	61143
19964	72330	84238	01371	10927	44948	35578	15037	58684	96445	85749	91777	28699	26744	21836	96211	37675	10108	32785	43794
08174	90940	91043	08409	67741	44708	43632	42794	76892	05620	04272	27961	63866	91498	05489	83112	12446	76399	93195	53714
84012	88636	07487	06479	56866	90485	74782	85521	70547	40113	94592	96221	77502	57556	58110	67452	20144	89819	91968	63596
53615	51681	27398	27407	60138	89963	88203	18776	30366	87627	30157	58464	00427	98880	69186	26402	68612	68618	08838	74939
57381	81250	22279	68993	02674	46255	77395	95424	69831	63786	30001	71279	22715	14060	34129	90218	15706	59650	53260	07758
23677	39818	21290	87394	44985	91827	49999	00722	35924	23334	56785	06711	86568	83918	67477	04960	01627	75406	25331	44061
90191	29983	78991	47125	15365	20033	60579	93508	60167	88076	97568	56237	78570	95255	54130	49029	27192	22018	41725	02357
12444	99118	70210	54269	45650	61384	91937	34743	24503	96626	77990	38402	38678	16868	09962	01587	90905	86549	42350	46991
90743	51955	10437	22544	51574	09678	29084	33602	59382	25780	73088	02738	55261	55197	20440	75620	32678	06244	48803	49099
82371	61231	68779	47156	13405	79324	95455	09528	05251	80101	23087	25877	89741	15817	04824	55889	71438	59675	44080	81313
43837	55029	88726	73952	33752	96641	61550	14060	91607	98322	92198	27240	61478	32528	92479	71651	99369	89519	18780	86812
21191	64174	77109	02480	63349	10917	04827	44122	82811	86632	44590	71457	87138	35123	48422	61380	07462	19140	04818	15238
66660	43133	34487	50679	03582	83828	35626	88083	23657	54820	68479	63954	63838	19532	17452	25026	82372	44136	32757	65875
60911	97836	53298	31206	67082	17149	31677	35643	40379	28972	43939	86744	13989	18554	16612	29573	93566	68612	65827	12346
96438	37712	28389	98040	19973	90780	61443	67541	56710	78463	40467	37024	03777	65347	81733	67084	84473	47020	56866	63615
81380	03692	25338	22099	09466	46959	19301	61626	09792	05087	42175	67030	65051	39542	86075	08061	59835	35754	10321	47095
08427	84610	56701	36773	97949	32024	20299	87077	31017	69258	20462	10702	21251	41204	29322	53043	17896	16267	04777	6115[illegible]
23597	93540	41470	84870	98546	54265	02772	05730	09003	33847	90533	42506	04119	50303	00017	04002	88789	29414	04603	3458[illegible]
99263	67501	35509	49427	50552	59158	16399	80523	19067	96107	84993	58089	66832	99297	68126	24423	14008	65703	34218	68094
55174	05064	48829	03920	73167	11307	69513	18922	96593	50901	86230	94810	55751	95603	05240	78716	38092	19164	43375	45148
63301	00091	59169	85856	24217	65636	24771	32898	16785	48246	29737	62495	30251	36036	34127	68366	45617	50770	31977	45753
49128	06433	17653	99959	94343	30811	84701	47158	71281	61493	94421	27661	42282	62909	95005	57469	81053	20661	00015	60295
78465	66161	93252	26941	20268	31159	50894	96715	13845	19588	32171	47982	74887	92618	51417	81997	90344	17285	59860	77272
20866	67768	04260	90308	75482	38033	45446	56630	56192	41308	37445	27546	68143	01548	77108	77728	01108	60043	25892	26225
94139	68285	28349	70455	71062	75770	14217	61565	26272	51534	07407	62540	51499	31989	49445	91064	14660	53430	53785	76709
86292	00498	64880	96114	48692	58603	47371	43636	59194	01396	27063	66851	38429	96928	69491	80517	25568	18508	29882	49549
54815	79606	31695	17658	74142	01597	98754	27342	80267	23452	48126	35691	57307	21315	37397	81041	62765	37150	78598	50415
47972	87663	12294	67113	48158	52941	88164	32825	04446	66927	81137	47449	48983	85064	37578	75073	76496	34514	86253	06383
39155	51456	90087	89195	53159	94462	94449	32352	48817	59990	71191	35755	93338	21217	06191	47718	50549	36632	21115	72229
20331	14850	24875	63303	11801	88056	85073	56984	15805	18118	71077	86539	53571	29601	43729	40865	27040	70219	24383	16729
03232	31567	91228	94194	86240	59403	90744	52321	67801	93818	71219	09215	54607	68444	57357	85595	13613	30424	22061	51356
45751	39372	70939	00970	72378	27101	24585	38376	78338	16102	33975	86854	89423	06960	91540	24998	79074	53461	31192	39638
52950	75475	80582	05625	95660	08177	43007	19174	68126	55955	02174	76709	22460	86674	77445	20875	60785	90623	34750	62709
83285	93480	06778	94561	69602	49439	28137	63495	65759	98474	85773	55399	09575	57313	20080	90408	30036	44649	22194	09934
09694	87305	47494	30121	61656	86750	73574	95558	82340	30398	98746	72975	45506	09577	36921	55919	54808	15514	03591	57071
29930	05702	71172	86252	84319	74133	12307	61788	67975	06784	26019	54367	60305	99034	07084	81464	60727	89554	95487	74214
07535	70621	21719	82521	92978	86978	69167	34625	61843	01754	54903	86411	15854	29504	56992	09056	36741	53903	09680	41471

Bryant Tuckerman
Thomas J. Watson Research Center
March 4, 1971

LET US ALWAYS STRIVE FOR ACCURCAY.

```
1 0 0 1 0 1 1 1 1 0 0 1 1 0 0 0 1 1 1 0 ——→ 620942)10
-1                 +1 ←—
————————————————————————                    Subtract M by
   1 0 1 1 1 1 0 1 0 1 0 0                  subtracting (M+1)
   1                     1                  and adding 1.
   ———————————————————————
     1 1 1 1 0 1 0 1 0 1 0 1
     1                     1
     ———————————————————————
       1 1 1 0 1 0 1 0 1 1 0 1
       1                     1
       ———————————————————————
         1 1 0 1 0 1 0 1 1 1 0 1
         1                     1
         ———————————————————————
           1 0 1 0 1 0 1 1 1 1 0 0
           1                     1
           ———————————————————————
               1 0 1 0 1 1 1 1 0 1 ————→ 701)10
```

For small numbers (say, from 9 to 20 digits), there are various schemes for determining primality, such as the one by Lehmer at the start of this article. For very large numbers of a special form, there is the Lucas-Lehmer test. In the mid range (20 to 30 digit numbers) there is still a need for new techniques. Two of the latest are reported in Mathematics of Computation, April 1974:

D. H. Lehmer and Emma Lehmer, "A New Factorization Technique Using Quadratic Forms."

R. Sherman Lehman, "Factoring Large Integers."

The latter article reports the factorization

$$2974231569940674843 7 = 372173423 \cdot 7991520281 9$$

in 122.6 seconds on a CDC 6400.

□

PROBLEM 65

Two Primes Problems

1

The 24 odd primes less than 100 are to be placed on the 24 faces of four cubes, in such a way that

1) Any toss of the four dice produces a sum that is divisible by 4; or

2) Any toss of the four dice produces a sum that is not divisible by 4.

Are either of these arrangements possible? If the 24 odd primes are taken to be those from 5 through 101, is either arrangement possible?

2

A succession of random numbers (uniformly distributed in the range from 001 to 999) is drawn. The numbers are progressively totalled until the sum is a prime number, at which time the game ends and the score is the number of numbers drawn. Problem: what is the distribution of the scores?

Casual calculation (by hand) indicates that the games are short. For 30 games, the distribution was as follows:

Length of game:	1	2	3	4	5	7	8	10	12	13	15	16
Occurences:	5	7	5	3	2	1	1	1	1	1	1	2

PROBLEM SOLUTION

Problem 59 (PC18-8) called for finding 100 digits of a number which, when squared, would have its 100 low-order digits the same as the original number. Such a number was named automorphic by Maurice Kraitchik, in his book Mathematical Recreations. In the 1942 edition of that book, the low order digits were given as

6259918212390625

but with an error in the digit indicated.

Sanford Goldfarb indicated a straightforward algorithm for extending the number: for any portion of the number that is known, calculate its square, as shown here. The known digits reappear in the product, and the digit to the left of those that repeat is the next digit sought.

```
            | 8 9 0 6 2 5
            | 8 9 0 6 2 5
            |------------
            | 3 9 0 6 2 5
      5 5 6 | 2 5 0
      5 5 6 | 2 5 0
7 9 2 1 0 0 |
------------|------------
7 9 3 2 1 2 | 8 9 0 6 2 5
```

Using just this method, a program was written for an IBM 1620 that produced the following result:

39530073191081698029385098900621665095808638110005
57423423230896109004106619977392256259918212890625

25 Squares

In the array of 25 squares shown at the right, the 25 primes less than 100 have been inserted on the east side of each square. The other 75 numbers from 00 to 99 are to be placed at the north, south, and west sides of the squares, with two simple restrictions:

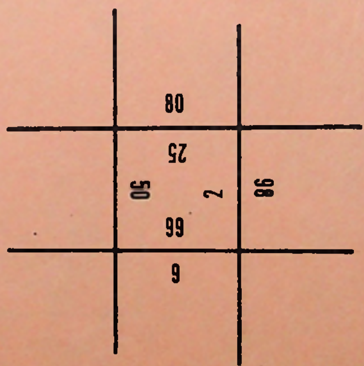
59	61	67	71	73
53	11	13	17	79
47	7	2	19	83
43	5	3	23	89
41	37	31	29	97

(1) The sum of the four numbers in each square is not to exceed 210, and

(2) The sum of the two numbers at each of the 40 common borders is not to exceed 105.

PROBLEM 64

Thus, the problem solution could begin with an arrangement like this one (the numbers have been placed facing outward in their squares):



There are a lot of arrangments that can be tried, and no solution is known. It may be that the constraints are too stringent to allow a solution at all. In that case, the Problem is: by how much would the parameters (210 and 105) have to be changed to permit a solution?

The following six prime numbers:

4068479
2034239
1017119
508559
254279
127139

Chained Primes

PROBLEM 67

are chained; that is, for each of the first five, (p-1)/2 is also a prime. The chain of 6 is the longest known, and Problem 67 is to find a chain of 7 or longer. There is no a priori reason to believe that longer chains exist, but it is likely. The simplest such chain is 47, 23, 11, 5, and 2

In searching for chained primes, it is not necessary to examine all consecutive primes. Table A shows the leading primes for chains of 3, which are quite abundant. It can be seen that the differences between successive entries are all multiples of 24, and that the entries themselves are all of the form 24K+23; it is not difficult to show that this is true. Given an entry value, the subsequent 24 integers can be expressed as the remainders modulo 24; that is, as numbers of the form 24K+M, and only those for which M = 1, 5, 7, 11, 13, 17, 19, or 23 could be primes. But since (p-1)/2 and (p-3)/4 must also be prime, the situation reduces to:

p =	1	5	7	11	13	17	19	23	mod 24
(p-1)/2 =	0	2	3	5	6	8	9	11	mod 12
(p-3)/4 =			1	2			4	5	mod 6

137279
143999
145007
146519
147047
148199
148727
149519
151967
157679
159119
165527
166487
166679
166919
167879
168527

A table of consecutive prime numbers, p, for which (p-1)/2 and (p-3)/4 are also primes.

A

and the only case that survives the sieve is 24K+23. This reasoning can be extended to chains of 4, 5, 6, or 7, leading to longer and longer jumps that can be made among the integers in searching for long chains. Thus, for chains of 4, the eligible numbers for the start of the chain must be of the form 48K+47; for chains of 7, the eligible numbers must be of the form 384K+383. □

More on Problem 43E (PC13-6) and the 44 terms shown in PC17-16. The problem, as stated:

> Take the integers from 3 to N. Circle the 3 and cross off every third remaining number. Circle the next remaining number (4) and cross off every third remaining number. Circle the next remaining number (5) and cross off every third remaining number. List the circled numbers, and give the 1000th circled number.

Sanford Goldfarb pointed out that the 44 terms given as a partial solution follow this pattern:

$$a_{n+1} = (3/2)a_n - 1/2 \quad \text{for } a_n \text{ odd}$$

$$a_{n+1} = (3/2)a_n - 1 \quad \text{for } a_n \text{ even}$$

(but that the 44th term given had a transposition error in the last two digits). Thus, the 1000th term would be approximately

$$(3/2)^{999} \text{ or, antilog } 175.9152$$

David Ferguson (of Group/3) analyzed the problem further, as follows:

"Suppose instead of taking the integers from 3 to N, one were to take the integers from 4 to N with all multiples of 3 omitted. Then clearly one would get the same circled numbers in the same order with the integer 3 missing. Consequently the number of integers eliminated preceeding the nth circled number (c_n) would be the same as the number of integers eliminated preceeding (c_{n-1}) in the original sequence. Hence the number of integers eliminated from c_{n-1} to c_n in the original sequence is exactly the same as the total number of integers preceeding c_n which were eliminated by 3. But since no c_n is divisible by 3 (except 3), this result can be stated as:

$$c_n - c_{n-1} = [c_n/3]$$

solving this for c_n there are two cases:

(1) $c_n = 3k+1$

(2) $c_n = 3k+2$

$k = (c_{n-1} - 2)/2$ or $3k+2 = (3c_{n-1} - 2)/2 = c_n$ (with c_{n-1} even)

$k = (c_{n-1} - 1)/2$ or $3k+1 = (3c_{n-1} - 1)/2 = c_n$ (with c_{n-1} odd)

Problem Solution

These last two results can be combined into:

$$c_n = \left[\frac{3c_{n-1} - 1}{2} \right]$$

Mr. Ferguson coded that solution for his System/3, yielding the first 1000 terms in 6 seconds of CPU time. The 1000th term is exactly

1334415470415079686859187725459711777188635805769080462194274578580740255136771951737068816443702825332078297437156879397892431324076228534214982057142873941052434787307231331 93

(177 digits), and the 44th term is 60540698. □

Problem Solution

Among the sieve problems (PC13-6) was that of Ulam, which calls for forming the sequence:

1, 2, 3, 4, 6, 8, 11, 13, 16, 18, 26,...

in which each new member can be formed in one and only one way by adding two different earlier numbers. The number 27 will not appear, since it can be formed in two ways (26 + 1, 16 + 11); the number 35 will not appear since it cannot be formed at all.

Associate Editor David Babcock found the 1000th term of the sequence to be 12294, using an assembly language program of 50 instructions. The first 110 terms (after the 26) are listed here to use as test data for students who wish to try extending the sequence.

28	36	38	47	48	53	57	62	69	72	77
82	87	97	99	102	106	114	126	131	138	145
148	155	175	177	180	182	189	197	206	209	219
221	236	238	241	243	253	258	260	273	282	309
316	319	324	339	341	356	358	363	370	382	390
400	402	409	412	414	429	431	434	441	451	456
483	485	497	502	522	524	544	546	566	568	585
602	605	607	612	624	627	646	668	673	685	688
690	695	720	722	732	734	739	751	781	783	798
800	820	847	849	861	864	866	891	893	905	927

The Future of Programmers

What jobs are there that rate the title "programmer"? The loose term can include any of the following:

1. Applications programmers. These are the people who write a new tape merge (in COBOL); who computerize the company's personnel file procedures; who maintain the payroll procedure in its weekly changes; who use Fortran to analyze a market survey; and so on. At the lowest level, these people are coders and, however skillful they may be at coding, that skill is not worth much in the marketplace. They include the users of the specialized languages (SIMSCRIPT, FORMAC, etc.).

2. Vendor system programmers. These are the people who create new versions of an operating system; who write the compilers (i.e., the translators); who create the support software for group 1. By definition, they work for the makers of the hardware. Their product may be bundled or unbundled, but traditionally there seems to be no pressure to make it satisfactory to its users.

3. User system programmers. These are the experts who are the interface between group 1 and the vendor-supplied software. They rectify the bugs found in the operating system and the compilers. They moderate the battles between the applications programmers and the vendor's representatives over the questions of whether the faulty output is hardware or software trouble and, if the latter, whose software. They may modify the vendor-supplied software to fit their organization's particular needs. They may also write systems software when the vendor's is inadequate and/or there is none in the marketplace that satisfies their organization's real or imagined needs.

4. Software company programmers. These are, or should be, the real pros who create the packaged and proprietary software for canonical tasks. They differ from the group 2 people in that their output must be of higher quality; that is, the software they produce must be responsive to the user's needs.

Group 2 programmers will be with us for a long time. A few hundred competent ones could probably satisfy the needs of the industry for the next decade, but we will have several thousand of them. Similarly, group 4 programmers will be needed for some time but, by definition, they are few in number.

Group 3 programmers will also be needed. Each installation will need one or more, particularly during the time when vendor-supplied software is as badly written and poorly documented as it is in 1974. When support software is created that is relatively error-free, human engineered, and idiot-proof, this group will dwindle.

Group 1 is presently the largest group. Applications programming is what students (in universities, community colleges, and trade schools) have in mind as their eventual work. We already have applications programmers by the tens of thousands, and we are busy creating thousands more each semester.

The thesis of this article is that we are training too many people for blind alley jobs; that the market for these programmers is already low and is getting lower. Three trends support this thesis:

1. Packaged programs. In the scientific/engineering domain, there are few problems that have not been programmed. These programs are far from perfect, and are not available in all languages for all machines, but there is little demand for a new program to invert a matrix, or to calculate standard functions. Even large problem situations (motor design, lens design, gear design, etc.) have been packaged, and more such situations will be covered. Although it is fun (and instructive) to write a new PL/I program for solving simultaneous equations by Gaussian elimination, there just is no market for the result; we've done that task, over and over.

In the business area, generating programs exist for all the stock jobs: sorting, report writing, file maintenance. Indeed, through large packages like Informatics' Mark IV, nearly every routine task of file manipulation can be performed without the necessity of writing a single instruction in the usual sense--and the various tasks of file manipulation cover 95% of everything business wants to do with computers.

2. It has been demonstrated (via systems like JOSS) that conversational computing can be a powerful tool to couple the man who has a problem directly with the machine, so that he can be his own programmer. The so-called interactive languages (e.g., on-line BASIC, or Fortran) are only a first approximation to conversational computing; that is, they are largely just remote job entry systems, and substitute an expensive terminal for an inexpensive keypunch. To be sure, they cut the turnaround time, but they are a long way from being conversational. True

conversational computing, which has existed up to now only in time-sharing mode, is very expensive, which is why it has almost completely disappeared. When conversational computing returns, in dedicated systems consisting of a mini computer and a terminal, the effective coupling of man and machine will again act to eliminate programmers as the middlemen.

3. There are problem situations in abundance for which turnkey systems--again centered around a mini--will appear. The paperwork problems of the operators of small businesses (doctors, lawyers, accountants, automobile salesmen, real estate salesmen, mailing list users, and so on) are sufficiently alike by types to be packaged.

These three trends add up to a bleak future for those who regard programming as a lifetime of work writing pretty routines. What the industrial world needs--pretty badly--is people who understand computing and data processing, and the writing of programs is the smallest part of that discipline. Of far greater importance is knowing what should be computed; how that computation can be performed efficiently and effectively; and--most important--how to verify that the computation is correct. If any of those elements is disregarded, the end result is not worth much, as, for example:

1 The correct results, done inefficiently; the hour run that could have been done in 10 seconds;

2 The correct results done ineffectively; the results are not presented in a meaningful way;

3 The correct results done by computer that could have been done analytically, or by looking up previous results;

4 Incorrect results, with the programmer unable to differentiate his computer-produced garbage from good work.

Sadly, the management of our industry seldom asks (or can tell) whether work has fallen into one of those four categories.

An even more serious problem is the tendency toward technological obsolescence among programmers. (A symposium in December 1973 on "Exploring the Future" dealt at some length with this topic.) The industry tends to force programmers into specialization, while raising their salaries year after year. Presently, a point is reached at which the individual finds himself out of date and replaceable by someone half his age at half his salary. That point now faces many men at age 50. Partly, the fault lies with the system of wages and responsibility we have engineered in our industry. Partly, it lies with the individual who has not kept up with his trade. In any event, it foretells trouble for many people, and again points up the dreary future for large numbers of programmers.

If the above analysis is even partly correct, then there will have to be serious revisions made in our training and education practices. The current emphasis is hundreds of courses is on the mechanics of programming (that is, the writing of instructions in some higher level language), usually on well-defined problems. This <u>training</u> may be necessary, but it is far from sufficient to satisfy the future needs of the computing world and, at its best, is not <u>education</u>.

--BY FJG

19

Log 19	1.27875360095282896153633347575692931795112933739450
Ln 19	2.94443897916644046000902743188785353723737926129913
$\sqrt{19}$	4.35889894354067355223698198385961565913700392523244
$\sqrt[3]{19}$	2.66840164872194486733962737197083033509587856918310
$\sqrt[5]{19}$	1.80198312731714230518255395296189025894370970228005
$\sqrt[7]{19}$	1.52292699821875339338696819152719299862141820457222
$\sqrt[10]{19}$	1.34237965096210479809378379300029234668987798576376
$\sqrt[100]{19}$	1.02988216191782398114014848587167216575078284094775
e^{19}	178482300.963187260844910033788722703883619733165166426195383201575692752239060170044051350752
π^{19}	2791563949.5978455652898978401209639170490063058647732336983964042328725477548397358909651344
$\tan^{-1} 19$	1.51821326518395490125645291855415597303504802920203
19^{100}	75051624198251984443456989853061891539043939434909537798332873934101480896578056472849915762891214746171016655874432115640378001